Proiect baze de date

Gestionarea biletelor de tren

[**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare. 2**](#_nbjcd2gzat2f)

[**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului. 3**](#_nbjcd2gzat2f)

[**3. Descrierea entităților, incluzând precizarea cheii primare. 3**](#_nbjcd2gzat2f)

[**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora. 3**](#_nbjcd2gzat2f)

[**5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor. 3**](#_nbjcd2gzat2f)

[**6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5. 3**](#_nbjcd2gzat2f)

[**7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. 3**](#_nbjcd2gzat2f)

[**8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7. 3**](#_nbjcd2gzat2f)

[**9. Realizarea normalizării până la forma normală 3 (FN1-FN3). 3**](#_nbjcd2gzat2f)

[**10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). 3**](#_nbjcd2gzat2f)

[**11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 3**](#_nbjcd2gzat2f)

[**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: 3**](#_nbjcd2gzat2f)

[**a) subcereri sincronizate în care intervin cel puțin 3 tabele 3**](#_nbjcd2gzat2f)

[**b) subcereri nesincronizate în clauza FROM 3**](#_nbjcd2gzat2f)

[**c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate 3**](#_nbjcd2gzat2f)

[**(în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri) 3**](#_nbjcd2gzat2f)

[**d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri) 3**](#_nbjcd2gzat2f)

[**e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE 3**](#_nbjcd2gzat2f)

[**f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH) 3**](#_nbjcd2gzat2f)

[**13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri. 3**](#_nbjcd2gzat2f)

[**14. Crearea unei vizualizări complexe. 3**](#_nbjcd2gzat2f)

[**15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n. 4**](#_nbjcd2gzat2f)

[**16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. 4**](#_nbjcd2gzat2f)

[**17. a. Realizarea normalizării BCNF, FN4, FN5. 4**](#_nbjcd2gzat2f)

[**b. Aplicarea denormalizării, justificând necesitatea acesteia. 4**](#_q0oeqr382)

# 

# **1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Modelul de date propus vizează gestionarea biletelor de tren pentru operatorii feroviari din România, inclusiv CFR. Scopul sistemului este de a facilita vânzarea și monitorizarea biletelor de călătorie emise pentru trenurile de pasageri care circulă pe teritoriul țării, precum și pentru a sprijinii companiile feroviare în analiza traficului și veniturilor.

Modelul oferă o bază pentru digitalizarea organizării traficului feroviar de călători și pentru crearea unor aplicații moderne de vânzare și rezervare bilete, compatibile cu nevoile pasagerilor și cu cerințele operatorilor de transport.

Utilitatea acestui model de date dată și de calculul automat al tarifelor în funcție de rută și serviciile dorite de călător, cât și de adaptabilitatea acestuia la adăugarea de noi rute și operatori.

Regulile de funcționare ce:

* Un călător poate cumpăra unul sau mai multe bilete.
* Un tren poate circula pe o singură rută.
* Tarifele aplicate biletelor variază în funcție de clasa de confort, distanță și eventuale promoții aplicabile.
* Fiecare vagon are o clasă de confort .
* Un călător poate beneficia de o singură promoție la un anume bilet, în baza unei legitimații.
* Trenurile opresc în mai multe stații din cadrul unei rute, dintr-o stație de plecare la una de sosire, memorându-se în ordine și eventuale stații intermediare.
* Trenurile aparțin unor operatori feroviari, iar un operator poate deține mai multe trenuri.

# 

# **2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**

Entitatea STAȚIE este identificată prin numele ei, acesta fiind unic (de ex. gările din două sate numite Săcuieni din România se numesc Săcuieni Roman și Săcuieni Bihor).

Sunt folosite chei primare artificiale pentru a identifica unic celelalte entități neasociative, de exemplu:

* Fiecare CĂLĂTOR are un identificator unic (id\_calator) și un cnp unic (nefolosit ca cheie).
* Fiecare TREN are un cod unic ca cheie primară, indiferent de ruta pe care circulă.
* Fiecare BILET este identificat prin id\_bilet, generat automat.

Chei externe:

* BILET trimite spre CĂLĂTOR, TREN, CLASA, TIPBILET, PROMOTIE ( deci are drept chei externe id\_calator, id\_tren, id\_clasa, id\_tipbilet, respectiv id\_promotie dacă are o promoție asociată).
* TREN are cheile externe: id\_operator, id\_ruta.
* VAGON are cheile externe: id\_tren, id\_clasa.
* LEGITIMAȚIE are cheile externe: id\_calator, id\_promotie.
* entitatea asociativă TARIF
* o relație de tip trei în schema entitate-relație
* entitate asociativă cu cheile externe: id\_ruta, id\_clasa, id\_tipbilet
* entitatea asociativă OPRESTE:
* folosește cheia primară compusă id\_rută + nume\_stație
* nume\_stație este cheia externă ce corespunde cheii primare din Stație

Constrângeri logice pentru menținerea validității datelor:

* Fiecare BILET este emis pentru o RUTĂ, la o dată din viitor în care trenul circulă.
* Un TREN este are o singură RUTĂ, dar o pe o RUTĂ pot circula mai multe trenuri.
* O STAȚIE apare o singură dată într-o RUTĂ.
* Un CĂLĂTOR poate avea cel mult o LEGITIMAȚIE pentru fiecare PROMOȚIE.

# **3. Descrierea entităților, incluzând precizarea cheii primare.**

1. CALATOR

* persoană ce poate cumpăra bilete de tren și opțional deține legitimații
* id\_calator#, nume, prenume, cnp, email

1. BILET

* titlu de călătorie, valabil pentru o anumită rută, clasă și tip
* id\_bilet#, id\_calator, id\_tren, id\_clasa, id\_tipbilet, id\_promotie(optional), data\_emitere, data\_calatorie, pret\_final

1. TIPBILET

* reprezintă clasificarea biletelor: dus, dus-întors, abonament etc.
* id\_tipbilet#, denumire, descriere

1. STATIE

* gară unde trenurile pot opri
* nume#, oraș, județ

1. RUTA

* secvență de stații cu orele la care se oprește trenul acolo
* id\_ruta#, data\_start, statie\_plecare, statie\_sosire

1. TREN

* tren ce circulă pe o anumită rută și operat de un anume operator
* id\_tren#, numar\_tren, categorie, id\_operator, id\_ruta

1. VAGON

* vagon a unui tren; are clasă de confort
* id\_vagon#, id\_tren, id\_clasa

1. CLASA

* nivelul de confort oferit de un vagon
* id\_clasa#, denumire, suprataxa

1. OPERATORFEROVIAR

* companie ce deține și operază trenuri
* id\_operator#, denumire, tip, cod\_inregistrare

1. PROMOTIE

* id\_promotie#, denumire, procent\_reducere, conditie

1. LEGITIMATIE

* document ce atestă dreptul unui călător la o promoție
* id\_legitimatie#, id\_calator, id\_promotie, emitent, valabil\_pana\_la

1. TARIF (entitate asociativă ternară)

* valoarea unui bilet pentru ruta, clasa și tipul biletului
* (id\_ruta, id\_clasa, id\_tipbilet)#, valoare\_tarif

1. OPRESTE (entitatea asociativă)

* stațiile incluse într-o rută, în ordine, cu orar
* (id\_ruta, nume\_statie)#

# **4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

* Relația cumpără: Un călător cumpără unul sau mai multe bilete; fiecare bilet este cumparat de exact un călător.
* Relația deține (între LEGITIMATIE și CALATOR): Un călător poate deține mai multe legitimații; o legitimație aparține unui singur călător.
* Relația corespunde: Fiecare legitimație este pentru o promoție; o promoție poate fi utilizată în mai multe legitimații.
* Relația are (între BILET și PROMOTIE): Un bilet poate folosi o promoție sau nu; o promoție poate fi folosită în mai multe bilete.
* Relația emis\_pt: Fiecare bilet este asociat unui tren; un tren poate avea mai multe bilete.
* Relația are (Între BILET și TIPBILET): Fiecare bilet are un tip definit; același tip poate fi folosit în multe bilete.
* Relația are (între BILET și CLASA): Fiecare bilet este emis pentru o clasă; o clasă poate fi folosită în multe bilete.
* Relația deține (între TREN și VAGON): Un operator deține mai multe trenuri; fiecare tren aparține unui singur operator.
* Relația circulă\_pe: Fiecare tren circulă pe exact o rută; o rută poate fi parcursă de mai multe trenuri.
* Relația compus\_din: Un tren este compus din vagoane; fiecare vagon aparține unui tren.
* Relația are (între VAGON și CLASA): Fiecare vagon este de o clasă; o clasă poate fi asociată cu mai multe vagoane.
* Relația opreste (entitate asociativă): O rută parcurge mai multe stații, în ordine. O stație poate apărea în mai multe rute.
* Relația tarif (relație ternară): Fiecare combinație unică de rută, clasă și tip bilet are un tarif definit.

# **5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.**

CALATOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_calator | NUMBER |  | Simplu | PK | NOT NULL |
| nume | VARCHAR2 | 50 | Simplu | - | NOT NULL |
| prenume | VARCHAR2 | 50 | Simplu | - | NOT NULL |
| cnp | CHAR | 13 | Simplu | - | NOT NULL, UNIQUE |
| email | VARCHAR2 | 100 | Simplu | - | NULL |

BILET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_bilet | NUMBER |  | Simplu | PK | NOT NULL |
| id\_calator | NUMBER |  | Simplu | FK | NOT NULL |
| id\_tren | NUMBER |  | Simplu | FK | NOT NULL |
| id\_clasa | NUMBER |  | Simplu | FK | NOT NULL |
| id\_tip\_bilet | NUMBER |  | Simplu | FK | NOT NULL |
| id\_promotie | NUMBER |  | Simplu | FK | NULL |
| data\_emitere | DATE |  | Simplu | - | NOT NULL, DEFAULT SYSDATE |
| data\_calatorie | DATE |  | Simplu | - | NOT NULL |
| pret\_final | NUMBER | (6,2) | Simplu | - | NOT NULL, > 0 |

TREN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_tren | NUMBER |  | Simplu | PK | NOT NULL |
| numar\_tren | VARCHAR2 | 10 | Simplu | - | NOT NULL, UNIQUE |
| categorie | VARCHAR2 | 20 | Simplu | - | NOT NULL, în (“IR”, “R”) |
| id\_operator | NUMBER |  | Simplu | FK | NOT NULL |
| id\_ruta | NUMBER |  | Simplu | FK | NOT NULL |

VAGON

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_vagon | NUMBER |  | Simplu | PK | NOT NULL |
| id\_tren | NUMBER |  | Simplu | FK | NOT NULL |
| id\_clasa | NUMBER |  | Simplu | FK | NOT NULL |
| numar\_vagon | NUMBER |  | Simplu | - | NOT NULL, >= 1 |

RUTA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_ruta | NUMBER |  | Simplu | PK | NOT NULL |
| data\_start | DATE |  | Simplu | - | NOT NULL |
| statie\_plecare | VARCHAR2 | 100 | Simplu | FK | NOT NULL |
| statie\_sosire | VARCHAR2 | 100 | Simplu | FK | NOT NULL |

STATIE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| nume | VARCHAR2 | 100 | Simplu | PK | NOT NULL |
| oras | VARCHAR2 | 100 | Simplu | - | NOT NULL |
| judet | VARCHAR2 | 100 | Simplu | - | NOT NULL |

OPERATORFEROVIAR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_operator\_feroviar | NUMBER |  | Simplu | PK | NOT NULL |
| denumire | VARCHAR2 | 100 | Simplu | - | NOT NULL, UNIQUE |
| tip | VARCHAR2 | 20 | Simplu | - | NOT NULL, IN ('public', 'privat') |
| cod\_inregistrare | VARCHAR2 | 30 | Simplu | - | NOT NULL, UNIQUE |

LEGITIMATIE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_legitimatie | NUMBER |  | Simplu | PK | NOT NULL |
| id\_calator | NUMBER |  | Simplu | FK | NOT NULL |
| id\_promotie | NUMBER |  | Simplu | FK | NOT NULL |
| emitent | VARCHAR2 | 100 | Simplu | - | NOT NULL |
| valabil\_pana\_la | DATE |  | Simplu | - | NOT NULL |
| (id\_calator, id\_promotie) | - |  | Compus | UNIQ | Unicitate pe combinația a două FK |

TIPBILET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_tip\_bilet | NUMBER |  | Simplu | PK | NOT NULL |
| denumire | VARCHAR2 | 30 | Simplu | - | NOT NULL, UNIQUE |
| descriere | VARCHAR2 | 200 | Simplu | - | NULL |

CLASA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_clasa | NUMBER |  | Simplu | PK | NOT NULL |
| denumire | VARCHAR2 | 30 | Simplu | - | NOT NULL, UNIQUE |
| suprataxa | NUMBER | (4,2) | Simplu | - | DEFAULT 0, CHECK (>= 0) |

PROMOTIE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_promotie | NUMBER |  | Simplu | PK | NOT NULL |
| denumire | VARCHAR2 | 100 | Simplu | - | NOT NULL, UNIQUE |
| procent\_reducere | NUMBER | (3,0) | Simplu | - | NOT NULL, BETWEEN 1 AND 100 |
| conditie | VARCHAR2 | 200 | Simplu | - | NULL |

TARIF

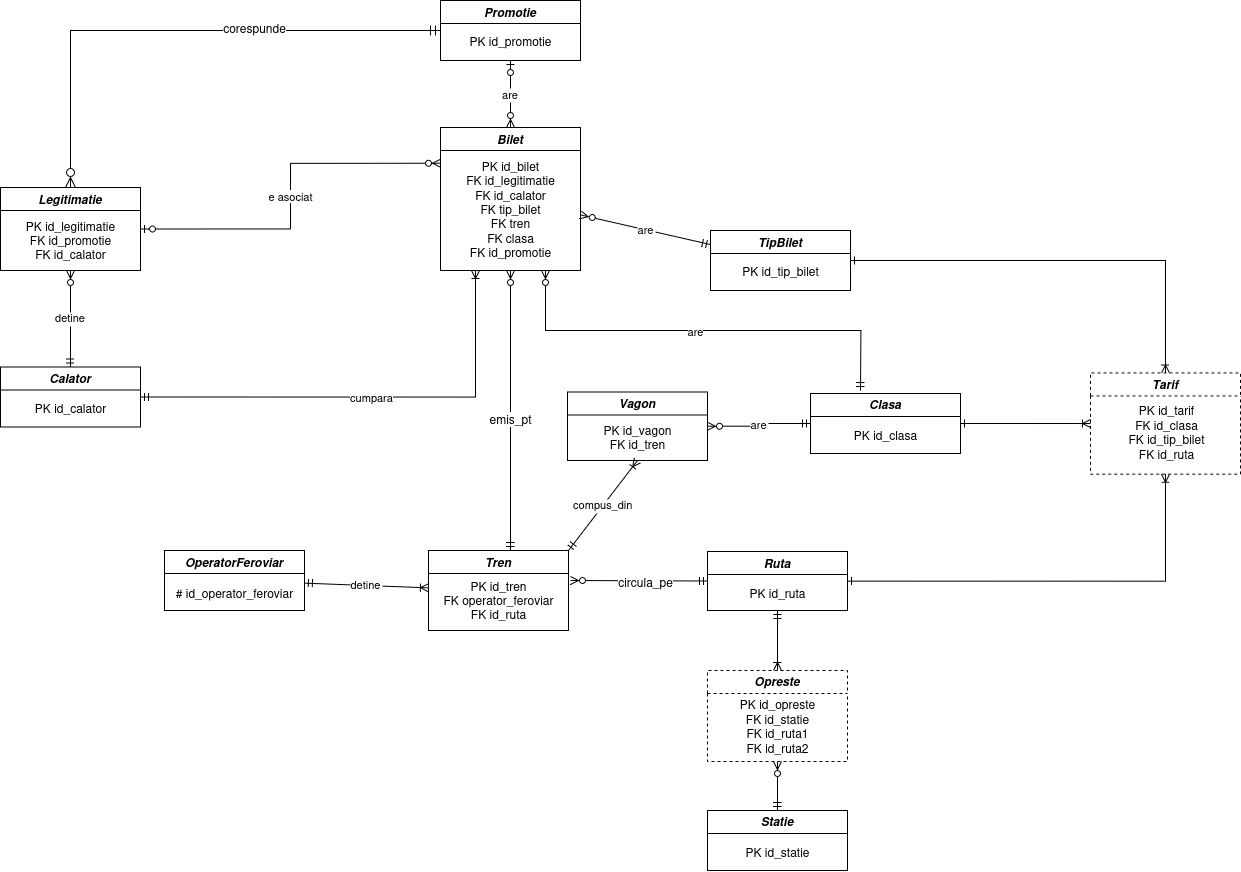
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_ruta | NUMBER |  | Simplu | FK | NOT NULL |
| id\_clasa | NUMBER |  | Simplu | FK | NOT NULL |
| id\_tip\_bilet | NUMBER |  | Simplu | FK | NOT NULL |
| valoare\_tarif | NUMBER | (6,2) | Simplu | - | NOT NULL, CHECK (> 0) |
| PK compus | - |  | Compus | PK | (id\_ruta, id\_clasa, id\_tip\_bilet) |

OPRESTE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Atribut** | **Tip** | **Dimensiune** | **Tip Atribut** | **Rol** | **Constrângeri** |
| id\_ruta | NUMBER |  | Simplu | FK | NOT NULL |
| nume\_statie | VARCHAR2 | 100 | Simplu | FK | NOT NULL |
| pozitie | NUMBER |  | Simplu | - | NOT NULL, >= 1 |
| ora\_sosire | DATE |  | Simplu | - | NULL (pentru prima stație) |
| ora\_plecare | DATE |  | Simplu | - | NULL (pentru ultima stație) |
| PK compus | - |  | Compus | PK | (id\_ruta, nume\_statie) |

# **6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.**

# **7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.**



# **8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.**

CĂLĂTOR(id\_calator, nume, prenume, cnp, email)

  PK: id\_calator

  UNIQUE: cnp

BILET(id\_bilet, id\_calator, id\_tren, id\_clasa, id\_tip\_bilet, id\_promotie, data\_emitere, data\_calatorie, pret\_final)

  PK: id\_bilet

  FK: id\_calator → CĂLĂTOR

  FK: id\_tren → TREN

  FK: id\_clasa → CLASA

  FK: id\_tip\_bilet → TIP\_BILET

  FK: id\_promotie → PROMOȚIE (opțional)

TREN(id\_tren, numar\_tren, categorie, id\_operator\_feroviar, id\_ruta)

  PK: id\_tren

  FK: id\_operator\_feroviar → OPERATOR\_FEROVIAR

  FK: id\_ruta → RUTA

VAGON(id\_vagon, id\_tren, id\_clasa, numar\_vagon)

  PK: id\_vagon

  FK: id\_tren → TREN

  FK: id\_clasa → CLASA

RUTA(id\_ruta, data\_start, statie\_plecare, statie\_sosire)

  PK: id\_ruta

  FK: statie\_plecare → STAȚIE(nume)

  FK: statie\_sosire → STAȚIE(nume)

STAȚIE(nume, oraș, județ)

  PK: nume (cheie naturală)

OPERATOR\_FEROVIAR(id\_operator\_feroviar, denumire, tip, cod\_inregistrare)

  PK: id\_operator\_feroviar

  UNIQUE: cod\_inregistrare

LEGITIMAȚIE(id\_legitimatie, id\_calator, id\_promotie, emitent, valabil\_pana\_la)

  PK: id\_legitimatie

  FK: id\_calator → CĂLĂTOR

  FK: id\_promotie → PROMOȚIE

  UNIQUE: (id\_calator, id\_promotie)

TIP\_BILET(id\_tip\_bilet, denumire, descriere)

  PK: id\_tip\_bilet

CLASA(id\_clasa, denumire, suprataxa)

  PK: id\_clasa

PROMOȚIE(id\_promotie, denumire, procent\_reducere, conditie)

  PK: id\_promotie

Entități asociative:

TARIF(id\_ruta, id\_clasa, id\_tip\_bilet, valoare\_tarif)

  PK: (id\_ruta, id\_clasa, id\_tip\_bilet)

  FK: id\_ruta → RUTA

  FK: id\_clasa → CLASA

  FK: id\_tip\_bilet → TIP\_BILET

PROGRAM\_TREN(id\_ruta, nume\_statie, pozitie, ora\_sosire, ora\_plecare)

  PK: (id\_ruta, nume\_statie)

  FK: id\_ruta → RUTA

  FK: nume\_statie → STAȚIE

# **9. Realizarea normalizării până la forma normală 3 (FN1-FN3).**

FN1 → FN2

Exemplu non-FN1:

TABEL: BILET\_NECORECT(id\_bilet, nume\_calator, prenume\_calator, tip\_bilet, clasa, pret)

Coloanele nume\_calator, prenume\_calator, tip\_bilet, clasa încalcă FN1 (sunt atribute repetitive).

Corectare FN1:

Separăm în tabele normale:

CĂLĂTOR: id\_calator, nume, prenume

BILET: id\_bilet, id\_calator, id\_tip\_bilet, id\_clasa, pret

FN2 → FN3

Exemplu non-FN2:

TABEL: PROGRAM\_TREN(id\_ruta, nume\_statie, pozitie, ora\_sosire, ora\_plecare, ora\_inceput\_ruta)

ora\_inceput\_ruta depinde doar de id\_ruta, nu de cheia compusă (id\_ruta, nume\_statie) → încălcare FN2.

Corectare FN2:

Mutăm ora\_inceput\_ruta într-un tabel separat:

RUTA(id\_ruta, ora\_inceput\_ruta, ...)

Exemplu de trecere FN2 → FN3

Tabel non-FN3:

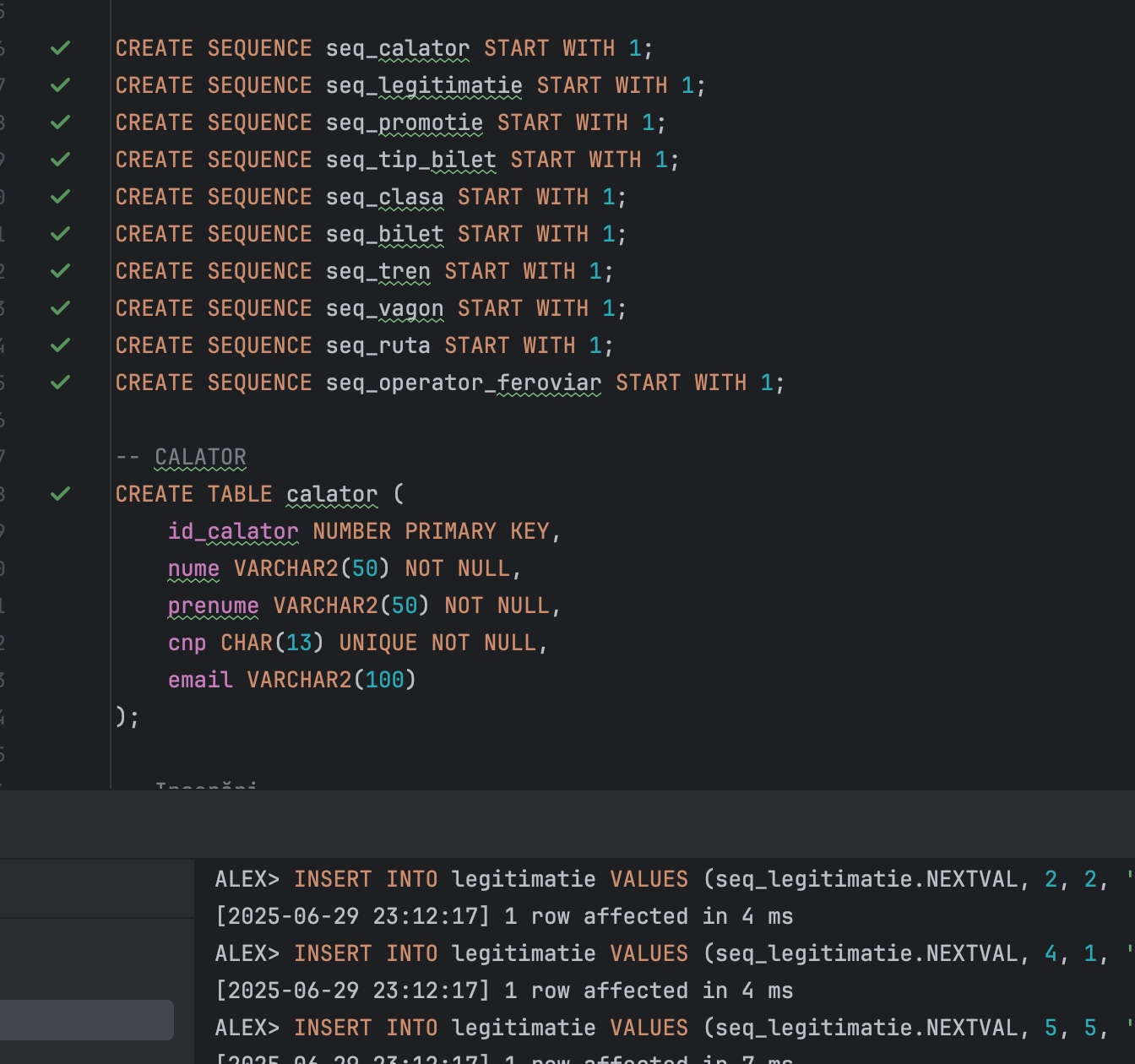
TABEL: VAGON(id\_vagon, id\_tren, id\_clasa, suprataxa)

suprataxa este dependentă de id\_clasa, nu de id\_vagon → încălcare FN3.

Corectare FN3:

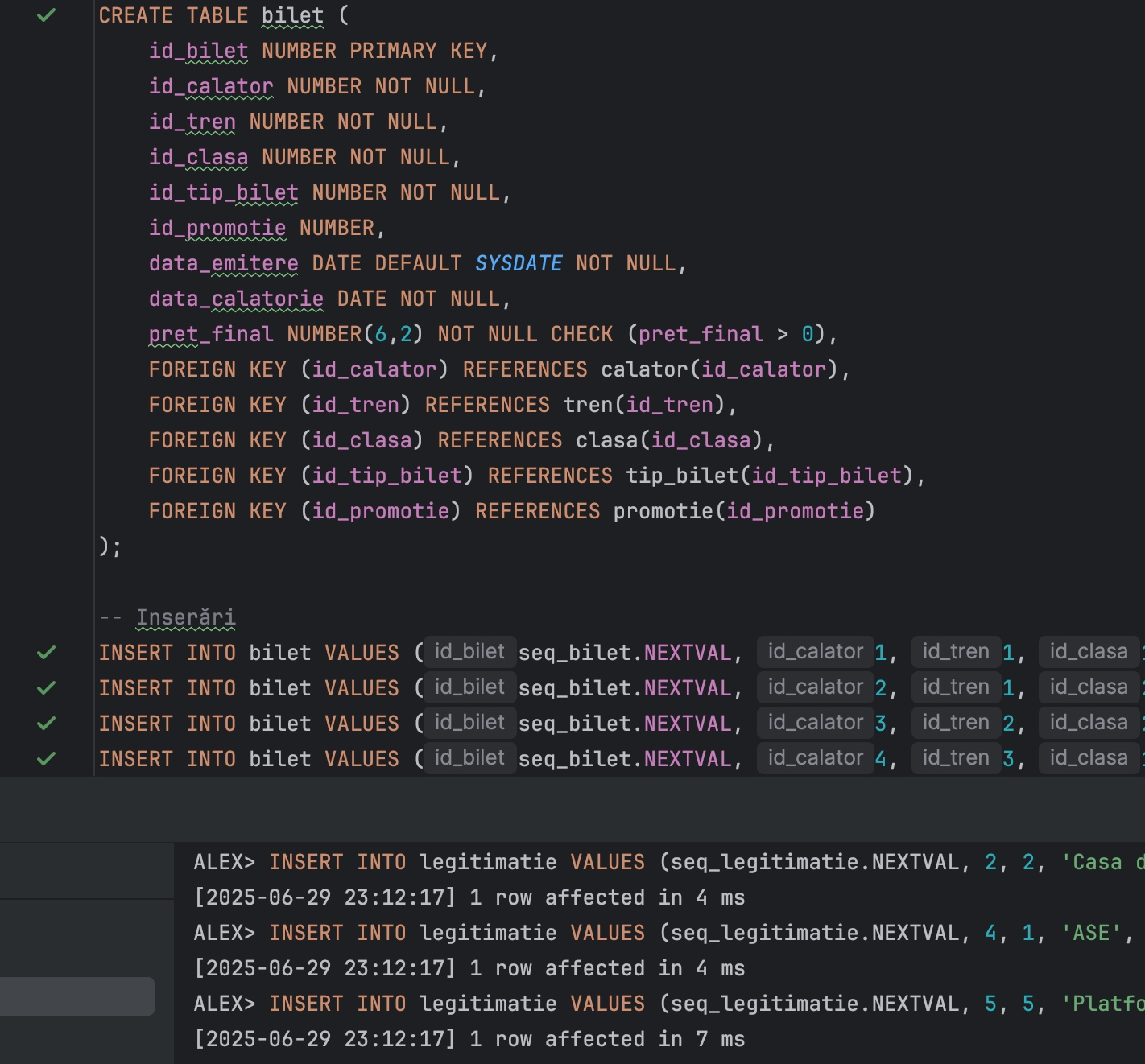
Mutăm suprataxa în CLASA, nu în VAGON.

# **10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).**



# 

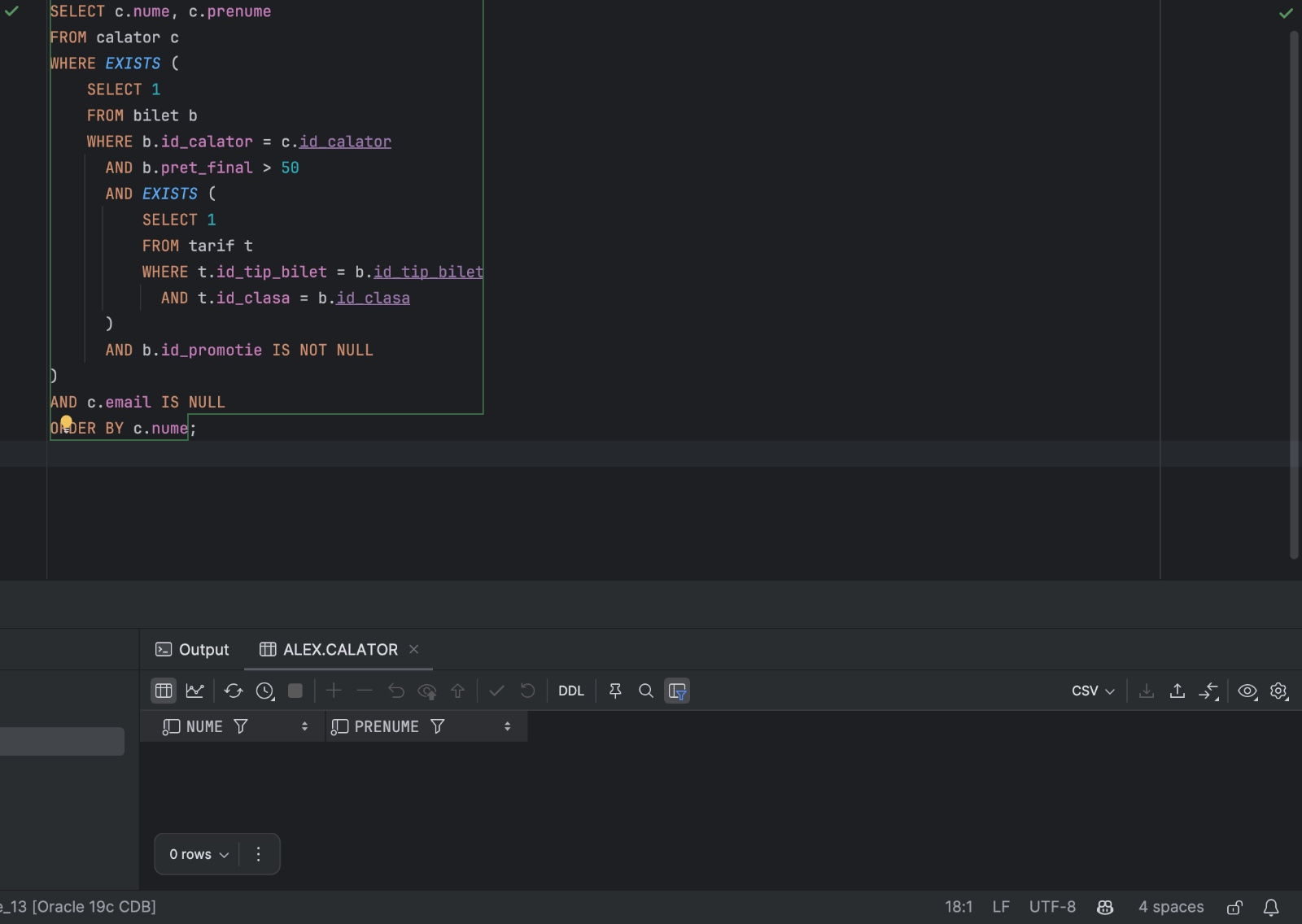
# **11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea**



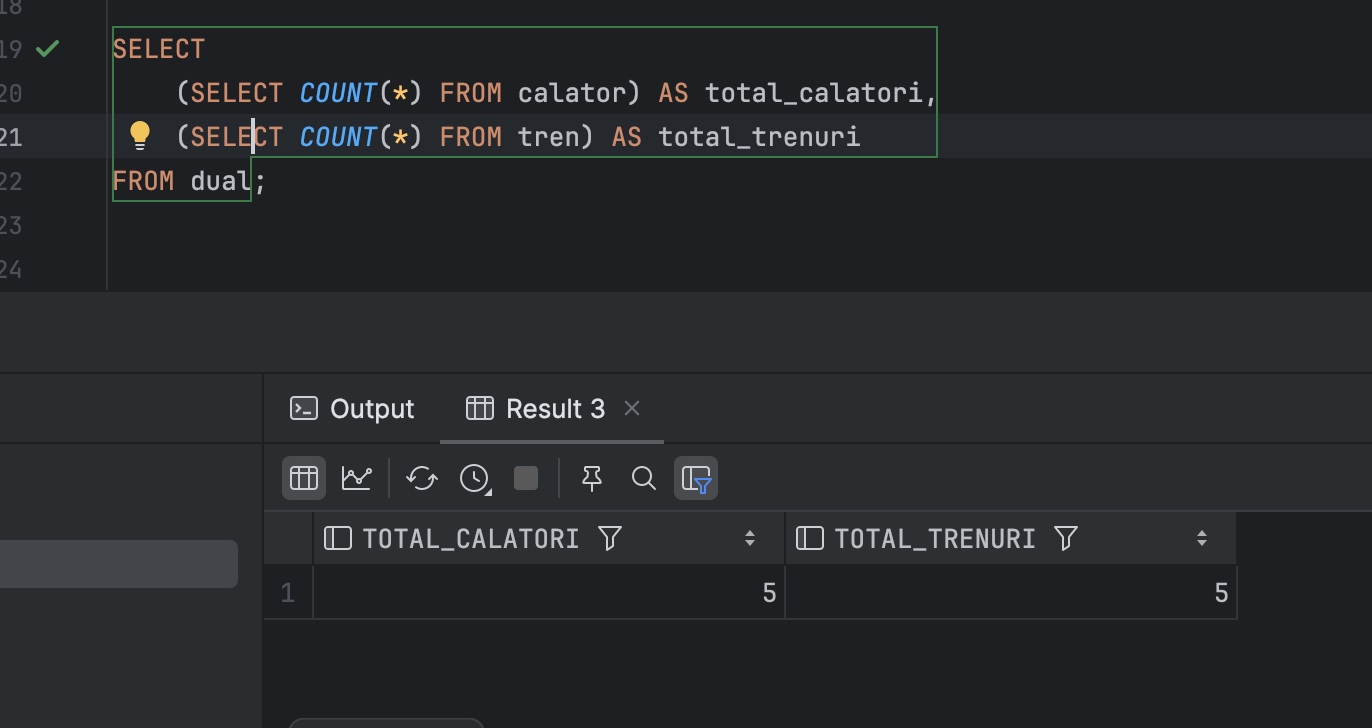
# **12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

# **a) subcereri sincronizate în care intervin cel puțin 3 tabele**

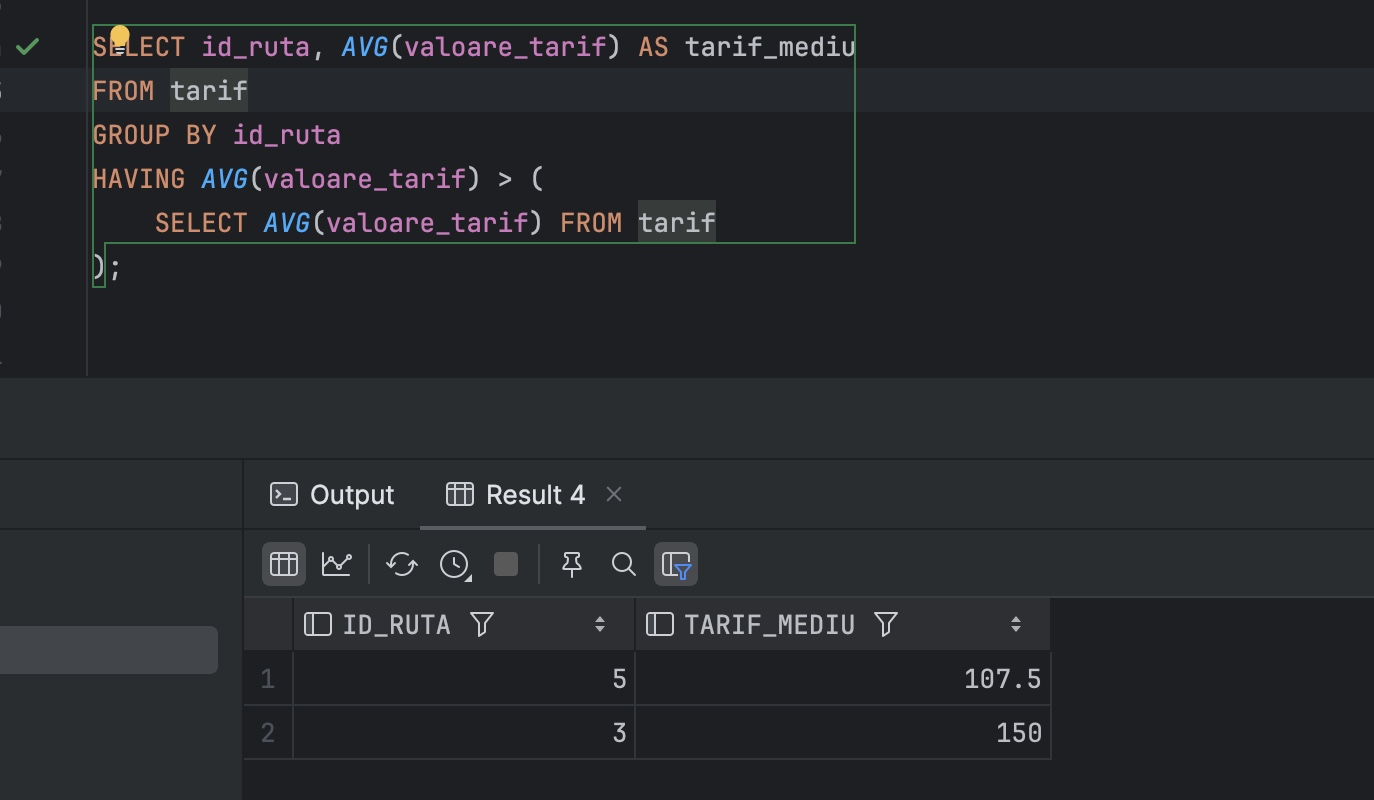
Se caută călătorii



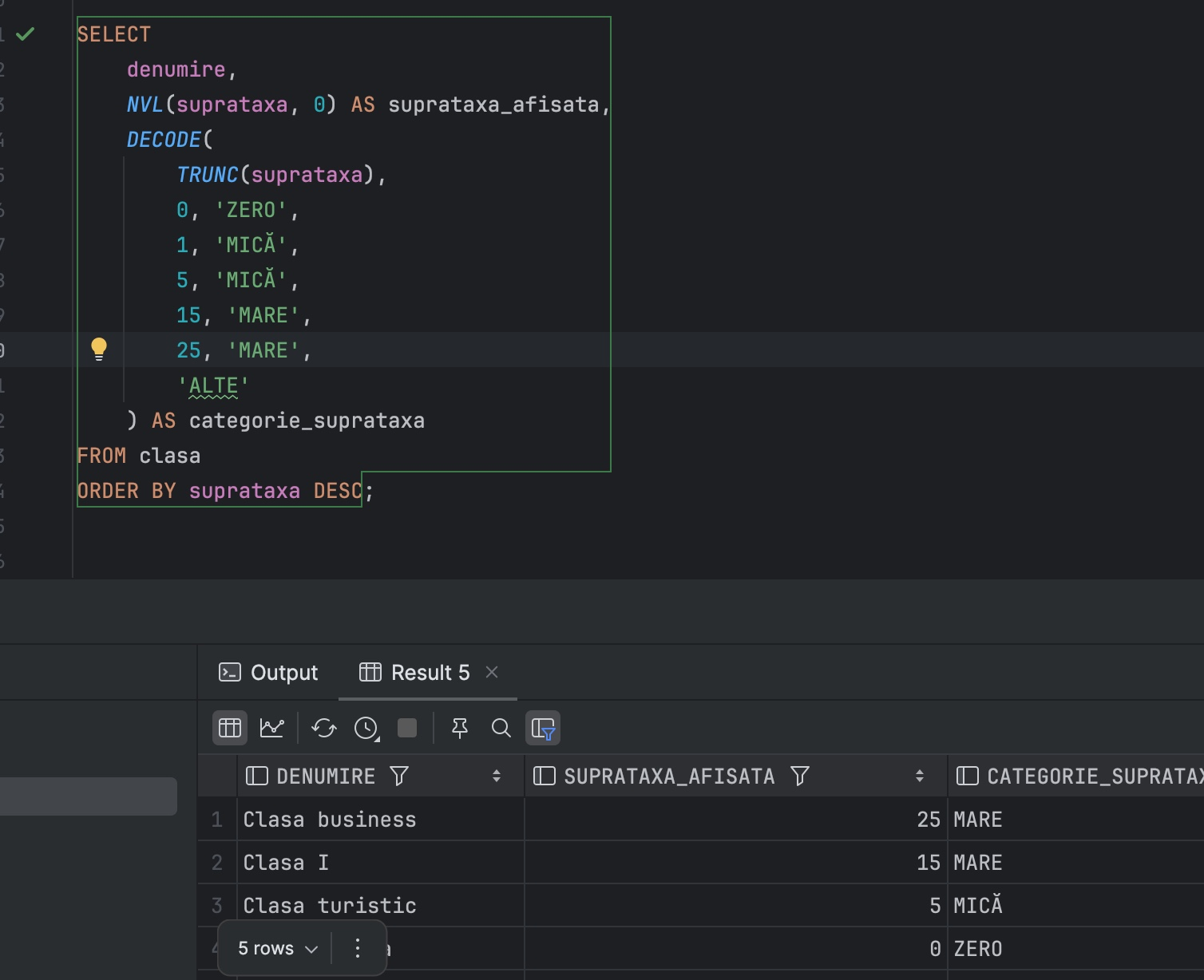
# **b) subcereri nesincronizate în clauza FROM**



# **c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate** **(în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)**



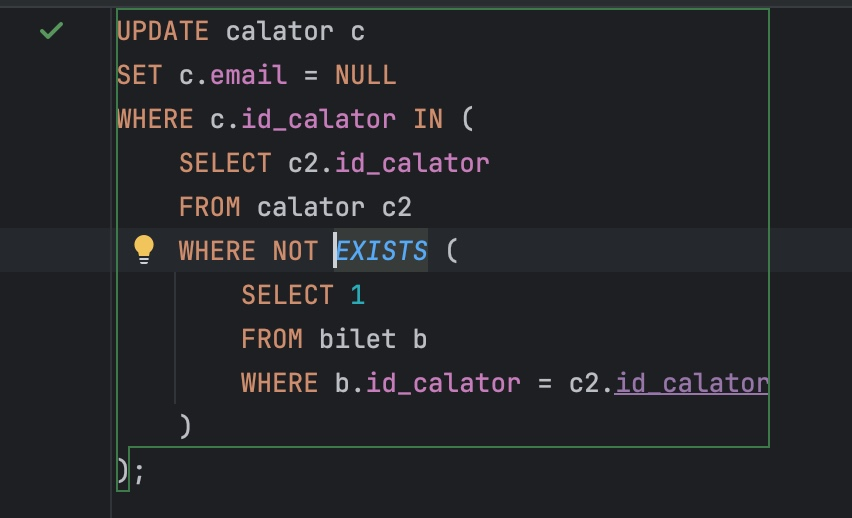
# **d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)**

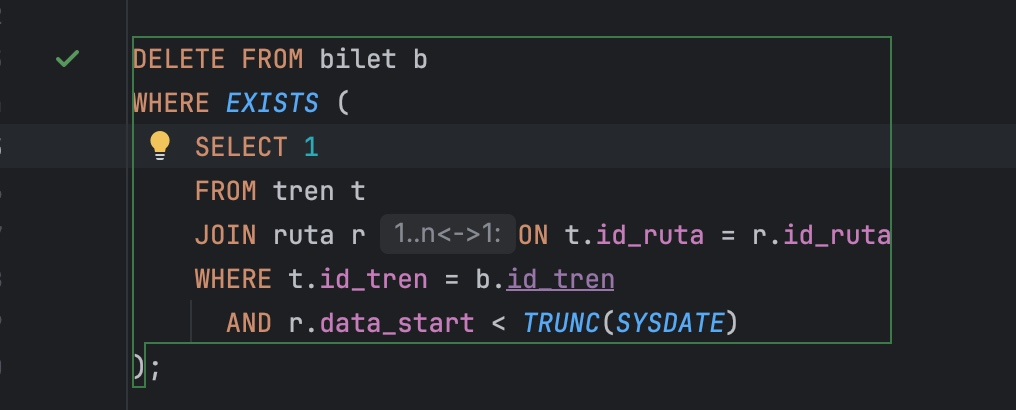


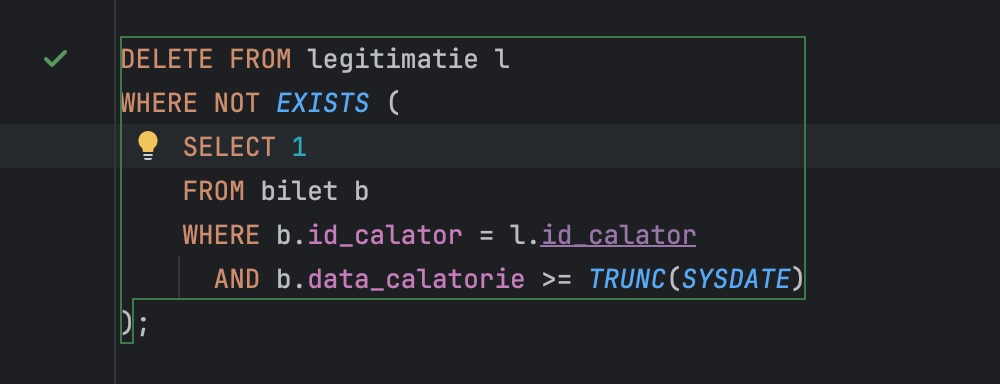
# **e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE și f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)**

# 

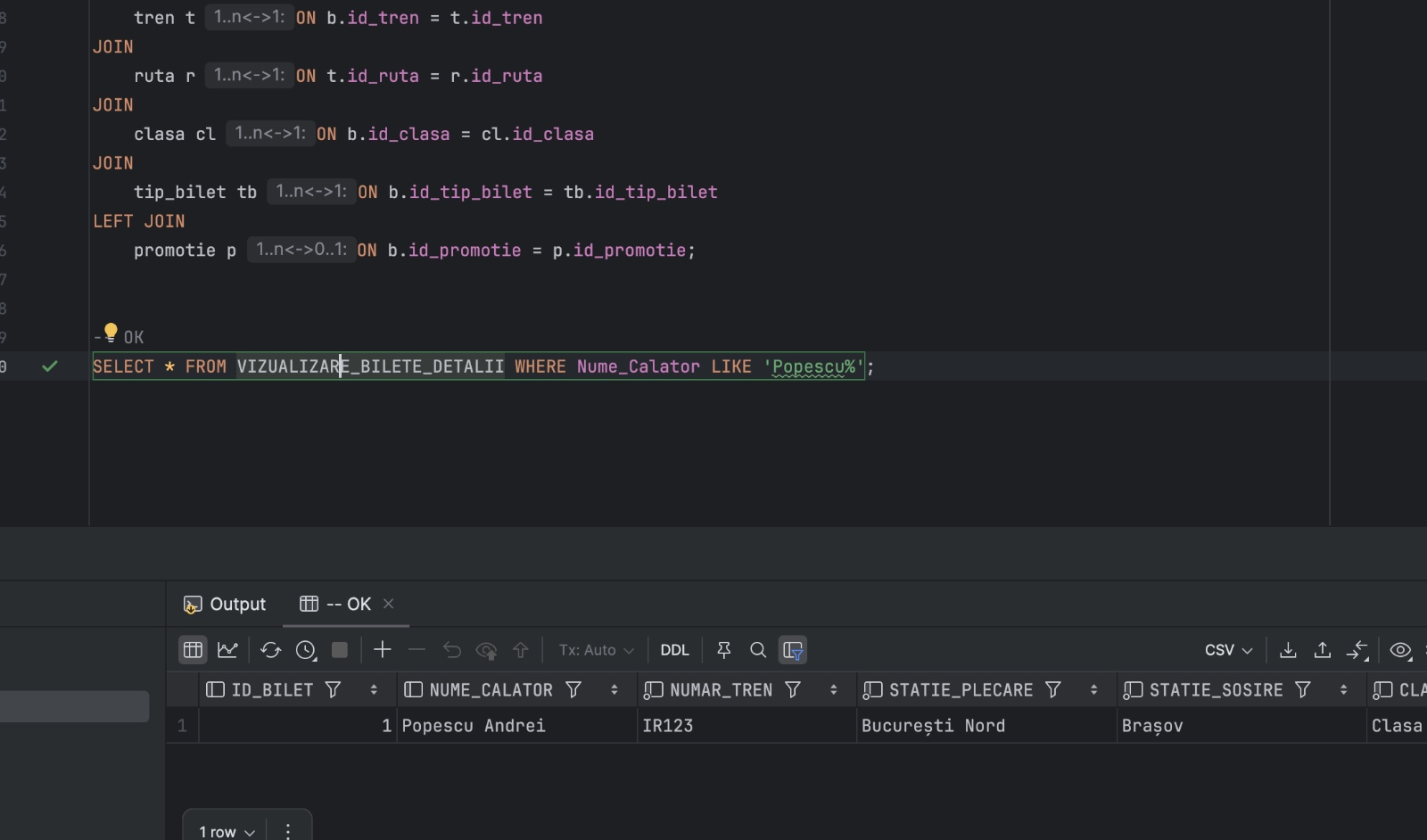
# **13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.**



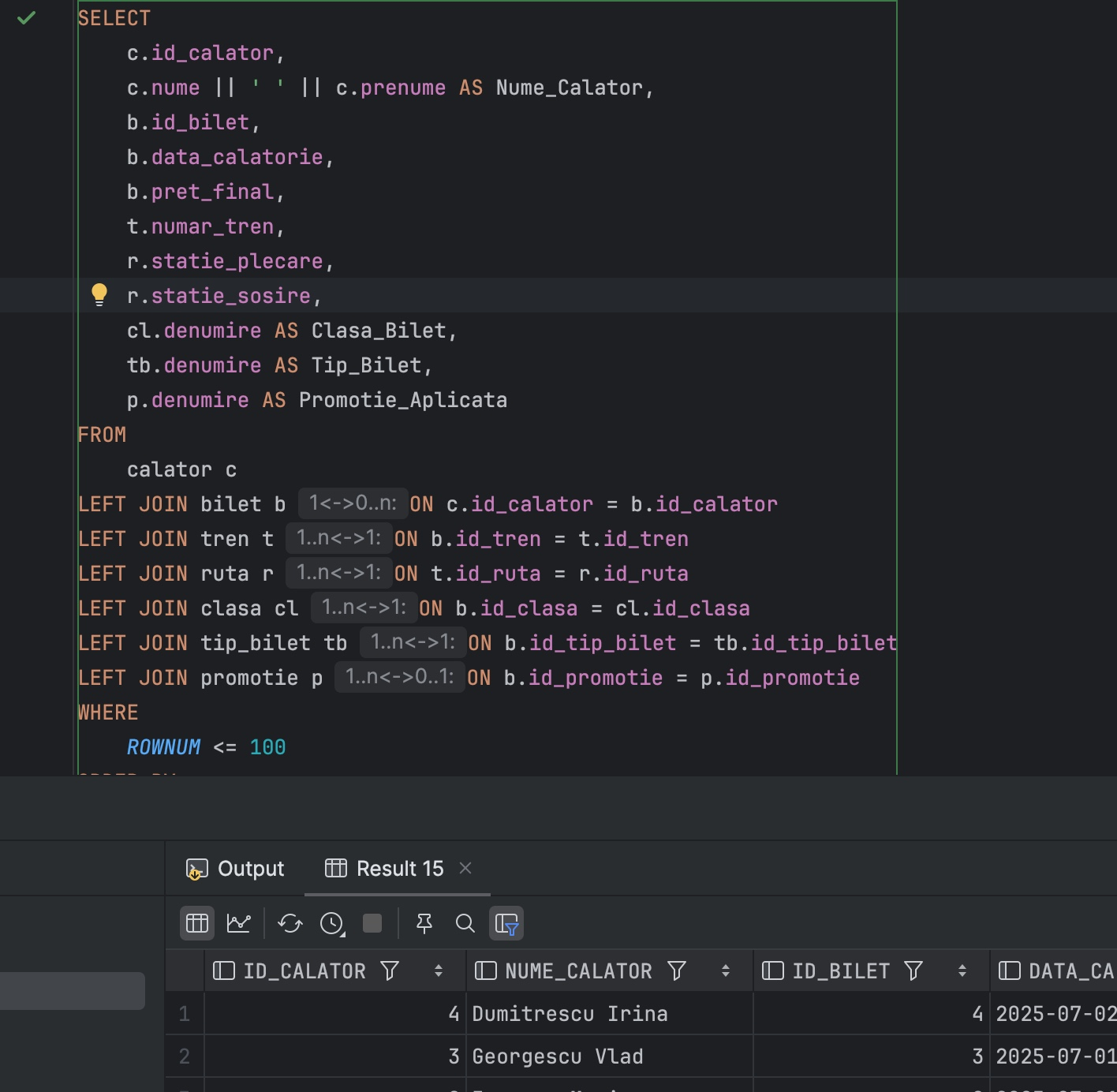


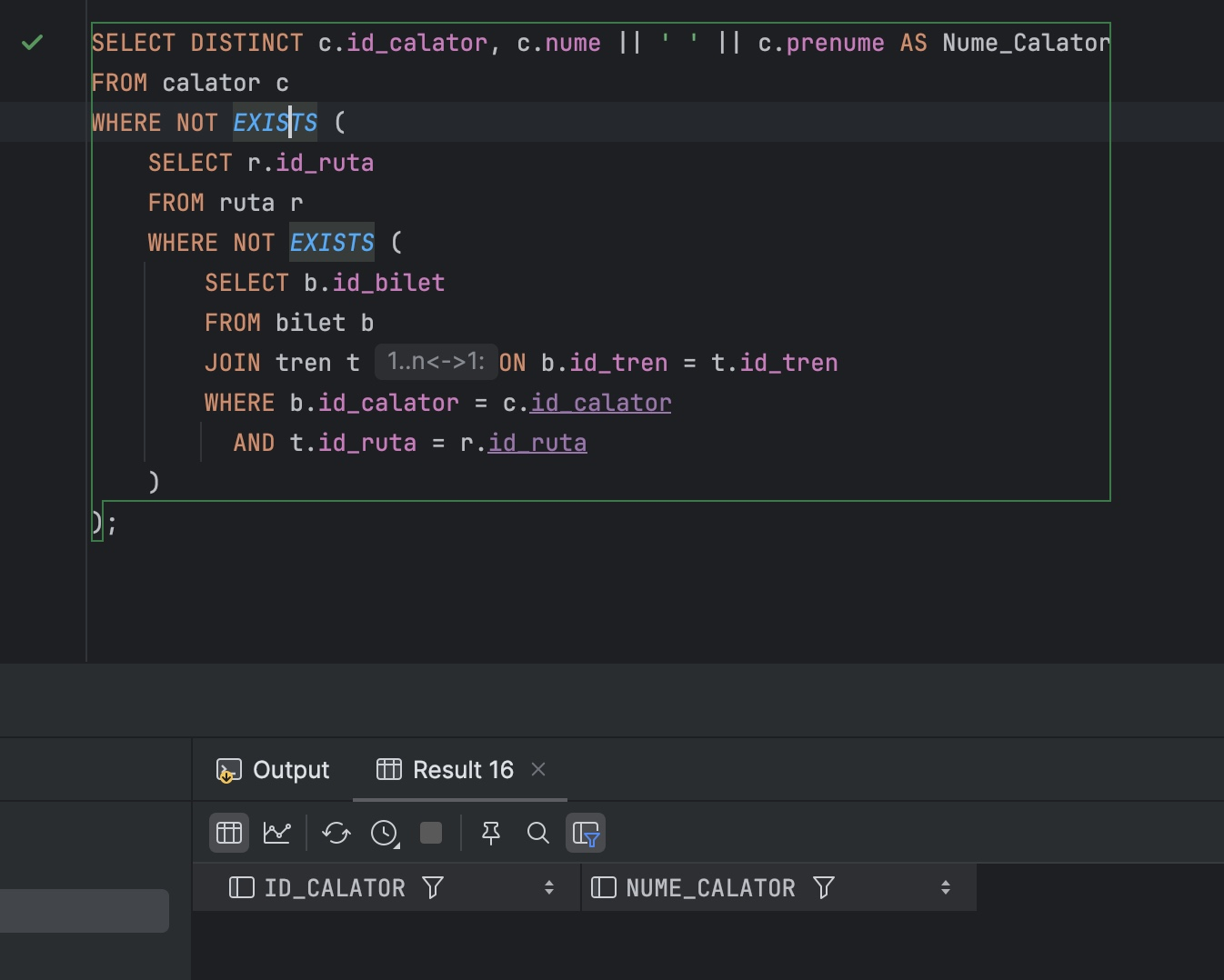


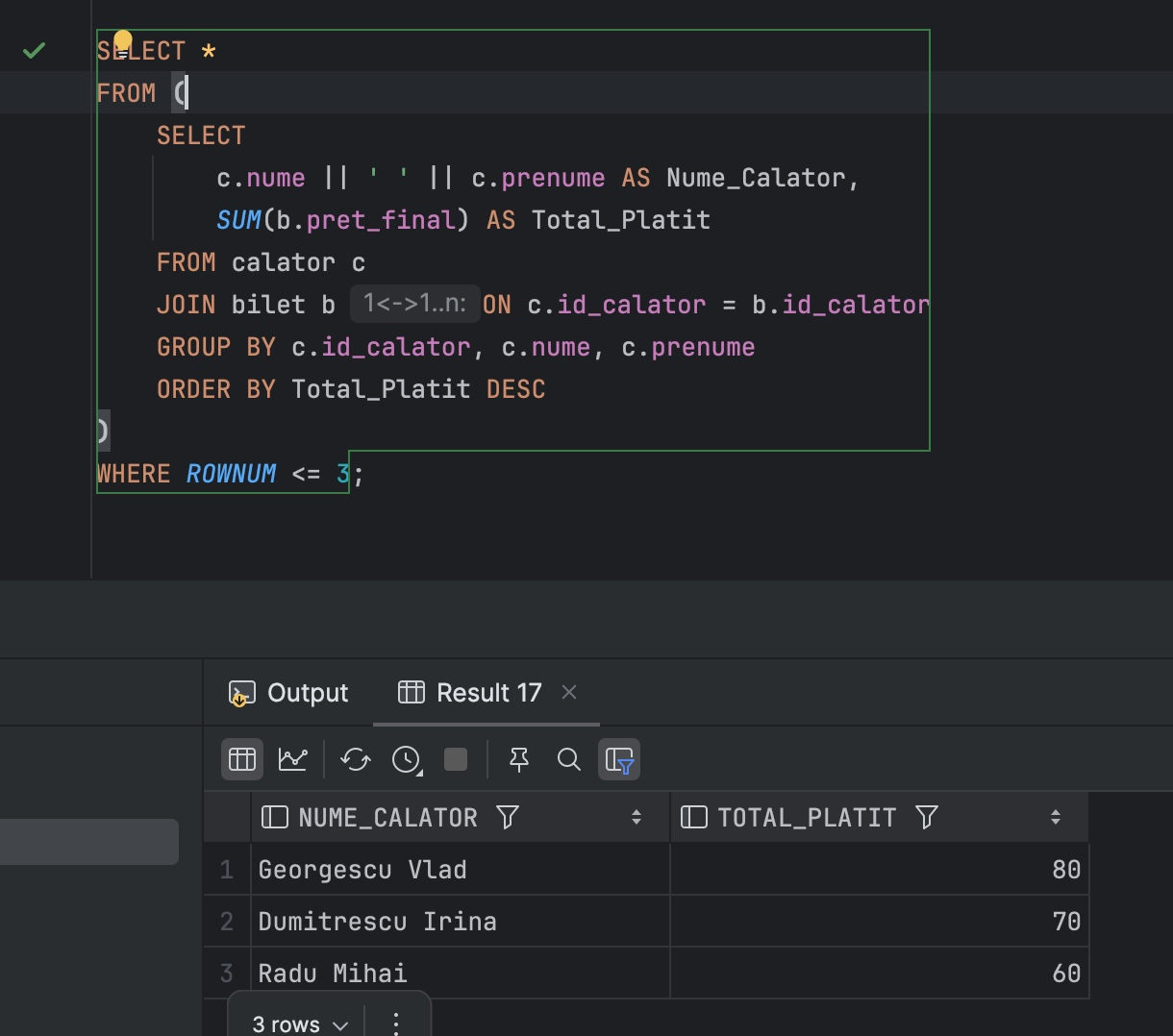
# **14. Crearea unei vizualizări complexe.**



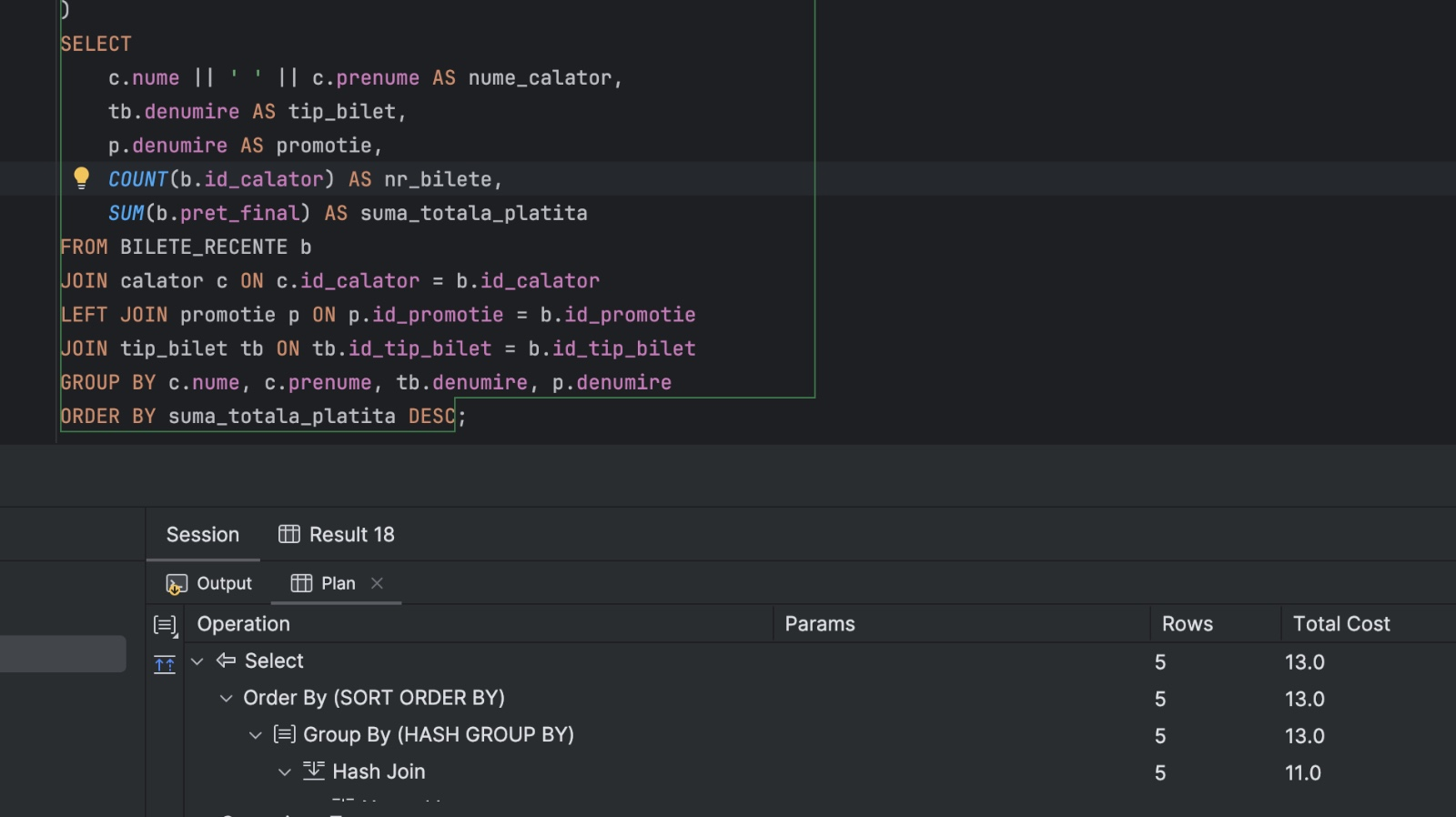
# **15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.**







# **16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale.**



# **17. a. Realizarea normalizării BCNF, FN4, FN5.**

Modelul de baze de date a fost proiectat inițial în FN3.

O relație este în BCNF dacă, pentru fiecare dependență funcțională netrivială XtoY, X este o supercheie (adică, X determină unic toate atributele din relație, chiar dacă nu este cheia primară). BCNF este o versiune mai strictă a FN3.

Majoritatea tabelelor din schemă utilizează chei primare artificiale (e.g., id\_calator, id\_tren). Acestea determină toate celelalte atribute din tabel, făcându-le automat superchei.

Dacă în tabela CALATOR am fi avut structură și dependențele astea:

CALATOR( underlinecnp, nume, prenume, email, cod\_legitimatie\_student)

Dependențe funcționale:

* cnp to nume, prenume, email, cod\_legitimatie\_student (PK determină tot)
* cod\_legitimatie\_student to cnp (Dacă am presupune că un cod\_legitimatie\_student este unic și identifică un singur cnp).

În cazul acesta, cod\_legitimatie\_student este un determinant, dar nu este o supercheie pentru întreaga tabelă (nu determină nume, prenume, email direct). Deși cnp este PK, existența unei chei candidate (cod\_legitimatie\_student) care determină PK (cnp) și este determinată de PK (cnp determină cod\_legitimatie\_student) introduce o încălcare a BCNF dacă cod\_legitimatie\_student nu ar fi cheie primară și ar determina un atribut non-cheie.

Corectare pentru BCNF :

Pentru a rezolva o astfel de situație, am descompune tabela astfel:

CALATOR(cnp, nume, prenume, email)

CALATOR\_LEGITIMATIE\_STUDENT(cod\_legitimatie\_student, cnp)

FK: cnp la CALATOR

# **b. Aplicarea denormalizării, justificând necesitatea acesteia.**

# 